DIMACS Technical Report 2009-13
April 2009

Global Ordering For Multi-dimensional Data: Comparison with K-means Clustering

by

Baiyang Liu
Dept. of Computer Science
Rutgers University
New Brunswick, New Jersey 08903

Casimir Kulikowski
Dept. of Computer Science
Rutgers University
New Brunswick, New Jersey 08903

Ilya Muchnik
DIMACS
Rutgers University
New Brunswick, New Jersey 08903

ABSTRACT

This paper describes a novel approach to estimate the quality of clustering based on finding a linear ordering for multi-dimensional data by which the clusters of the data fall into intervals on the ordering scale. This permits assessing the result of such local clustering methods like K-means so as to filter inhomogeneous or outlier clusters that can be produced. Preliminary results reported here indicate that the method is valuable to determine, in two dimensions, the number of visually perceived clusters generated by a mixture of Gaussian distribution model, corresponding to the number of actual generating distributions when the means are far apart, but corresponding to the reduced number of clusters arising from the perceived admixture of overlapping distributions when means are chosen to be close.

1 Introduction

This paper presents a new type of ordering for multi-dimensional data which can help eliminate outliers for clustering results such as those generated by K-means procedures. It is based on the notion that goodness of clustering is related to finding similar solutions resulting from very different methods. Consistency of results from different methods is considered as an estimate of the stability of a result.

The Ordering Procedure which we have developed is taken from the preprocessing algorithms developed for optimization of so-called Quasi-Concave set functions. The motivation came our observation that K-means clustering without outliers corresponds to a partition of continuous fragments in an ordering, such that the K-means clusters are associated with the fragments. If such an observation proves to be reproducible, one can then map the results from a K-means clustering onto the ordering sequence and estimate their consistency. This consistency can then be considered as a measure of quality for the clustering.

The paper is organized into 7 parts. Section 2 presents Quasi-Concave Set Functions and our ordering procedure (OP) taken from the optimization algorithm. A short introduction to clustering and K-means is given in Section 3 and experiments on synthetic mixture-of-Gaussian models are described in Section 4, including systematic performance of the K-means procedure on these models for different K, ordering each set of data with direct comparisons to visually detected and generating clustered distributions. Section 5 proposes a method for estimating clustering stability. Experimental results are presented in Section 6, with conclusions and future work in Section 7.

2 Ordered Sequence by Quasi-Concave Set Function

This section describes a procedure for ordering multi-dimensional objects into a linear sequence that preserves global similarity. This method is taken from the algorithm for optimizing Quasi-Concave Set Function, which has been applied in the areas of bioinformatics [5], image segmentation [6], gait recognition [4], and others. Quasi-Concave Set Functions are introduced first, followed by a description of the procedure with a simple illustrative example.

2.1 Quasi-Concave Set Function

Let Q denote a set of objects in r dimensional space.

$$Q = q_i, q_i \in \Re^r \ i = 1, ..., n \tag{1}$$

And H denotes a subset of Q, with $F(H)$ for measuring the proximity among elements in set H. An optimal subset, or the densest cluster H^*, can be defined as the subset with the largest $F(H^*)$.

$$H^* = argmax_{H \subseteq Q} F(H) \tag{2}$$

The optimal solution of (3) can be found efficiently when $F(H)$ is Quasi-Concave[5],

$$F(H_1 \bigcup H_2) \geq \min\left(F(H_1), F(H_2)\right) \forall H_1, H_2 \subseteq Q \tag{3}$$

In this work, $F(H)$ is defined as

$$F(H) = \min_{i \in H} \pi(i, H) \tag{4}$$

with linkage function $\pi(i, H)$ designed to measure the similarity of $q_i \in H$ to all other points in current set H. Let d_{ij} as the Euclidean distance between q_i and q_j, the similarity linkage function is defined as

$$\pi(i, H) = \sum_{j \in H} e^{-d_{ij}/(\sqrt{2}\sigma)} \tag{5}$$

Thus, H^* is the subset of Q with maximum of the least similarities inside each subset. $F(H)$ is Quasi-Concave if the linkage function $\pi(i, H)$ is monotonically increasing[5], $\pi(i, H) \geq \pi(i, H'), \forall H' \subseteq H \subseteq Q$

It is easy to show that linkage function (6) is monotone increasing. This guarantees an efficient algorithm for computing the optimal solution.

2.2 Ordered Sequence

Let $m(H)$ denotes an element $i \in H$ that reaches $F(H)$, that $m(H) = argmin_{i \in H} \pi(i, H)$. If more than one data points reach $F(H)$, one of them can be randomly chosen for $m(H)$. Given Q as initial H, ordered sequence M can be generated by repeatedly removing $m(H_t)$ from H_t and find $m(H_{t+1})$ until H_t becomes empty.

$$M = \{m(H_i) | i = 0, ..., n-1\} \tag{6}$$

The Procedure:

Input: n data points in r dimension

Output: Ordered sequence M with length n

1. Let $t = 0$; $H_t = Q$; $M = \emptyset$
2. Find $m(H_t) = argmin_{i \in H_t} \pi(i, H_t)$
3. $M = M \bigcup m(H_t); H_{t+1} = H_t - m(H_t)$
4. $t = t + 1$ and repeat from step 2 until $H_t = \emptyset$

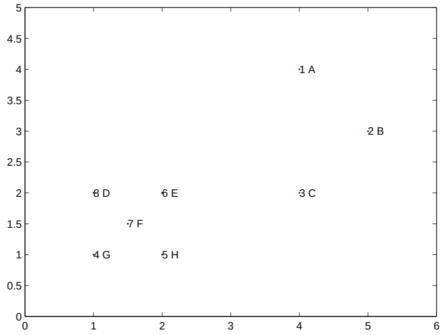


Figure 1: Simple example of ordering sequence

This procedure takes the whole dataset as input and returns a sequence of ordered data points. It is the preprocessing procedure for finding an optimal cluster H^* that is most compact. The sequence scans this dataset from the point that is least related to the set, to that which is the most related point, since the closest similar point is chosen at each step. Only the resulting ordered sequence M is of interest for present purposes in this paper.

Let us consider a simple example as shown in Figure 1 to demonstrate the procedure. There are 8 data points A(4,4), B(5,3), C(4,2), D(1,2), E(2,2), F(1.5,1.5), G(1,1), H(2,1). We consider them as vertices in a fully connected graph. Based on the procedure and linkage function in (6), A is chosen in the first iteration since it is the furthest point with smallest linkage value of 2.703, and removed from the current set. B is selected out of the remaining seven points in the second iteration with linkage value of 2.23. C,G,H,E,F,D are removed from the subset gradually in steps until the set is empty or the linkage function becomes zero. The resulting sequence is A,B,C,G,H,E,F,D ordered by similarity to the set. From this simple example, we observe that the points in the ordered sequence generally start in one sparse cluster (A,B,C), and then move to another denser one (G,H,E,F,D) when using similarity as linkage function. After removing the first three points, only one cluster is left in the set. The sequence moves from the outer layer to the inner layer in this one cluster case. A similar outcome has been observed for multi-dimensional data.

Our procedure orders any dimensional data onto one sequence which preserves its grouping structure by placing points from the same compact group into intervals on a line representing the measure of compactness, or fragments of the sequence. If this observation is reproducible, by calculating the variance of neighboring points for each point in the sequence order, one would expect to see spikes at the group boundaries for those data points or interval boundaries in the sequence, in contrast to much smaller variance for the points inside a compact group, as will be shown in Section 6. Thus, a raw data partition (into sequence intervals) can be obtained by applying a threshold to the variance values. Even with only sequence instead of any knowledge of the intervals, we can evaluate clustering results by analyzing the consistency between the ordering and visually perceived clustering in simple two dimensional examples. K-means is compared in the paper with the different number of clusters generating the samples and with the visually perceived clusters when the generating distributions overlap. A method for estimating the number of clusters K for the K-means procedure can thus be obtained, and is discussed next.

3 Global Ordering for Estimating the Number of Clusters in K-means

There are two types of clustering methods which minimize average variance for similarity-based data partitions: those which apply a 1) global search for the partition, usually with different types of genetic optimization algorithms [5], and 2) a search for a local optimization for the same criterion over partitions with a predefined number of clusters. All procedures of the last type are called K-means algorithms [1]. Because K-means procedures are very simple in practice, they can be applied many times to the same data to form different initial partitions, yielding a reasonably good estimate of the minimum of a criterion.

Unfortunately, there are no good ways of estimating how close a minimum reached by such a local method is to the global minimum solution. From this perspective, practitioners use a lot of post-processing heuristics to estimate how good a clustering is. The heuristics can be divided into two groups. One is related to the use of additional information which can be correlated with cluster extraction, and providesan informal estimate. The second one focuses on building additional numerical criteria-heuristics for the estimation. Usually they are based on an idea that "good clustering" has to be stable. However, even though many such heuristics exist, practitioners are always looking for new, more generalizable ones.

The K-means method [1] is a simple and fast iterative algorithm for locally searching for K centroids of the data such that they minimize the total variance of a dataset:

$$V = \sum_{i=1}^{K} \sum_{x \in C_i} (x - \mu_i)^2 \tag{7}$$

Every data point is assigned to one of the K clusters, the center of which is the nearest one. The local search solution is as follows.

1. Arbitrarily or heuristically select K points as initial centers μ_i .

2. For each $i \in \{1, .., K\}$, let C_i denotes the set of data points that are closer to μ_i than they are to μ_j for any $j \neq i$.

3. For each $i \in \{1, .., K\}$, set μ_i as the mass center of all data points in C_i; $\mu_i = \frac{1}{|C_i|} \sum_{x \in C_i} x$.

4. Repeat to Setp 2 until C not longer change.

K-means is attractive for its simplicity and efficiency. However, as a local procedure, it does not always result in clustering that can be confirmed by visual perception, often leading to cluster structures that are too sensitive to small changes in the criterion function [7]. In this paper, it is stated that a good clustering should be stable based on two different principles: (1) Stability to a small perturbation (Perturbation either in the data or in the basic procedure or in both), (2) similarity of results with other results produced by very different clustering procedures. Our ordering procedure is a totally different procedure working with local clustering methods, which is consistent with the clusters' order. Data points within one cluster form an continuous fragment on the ordering sequence. But there will be much more perturbation if the clustering is incorrect. It is possible to validate a clustering result by analyzing the similarity (Stability) between the clustering and the global ordering as discussed later. With a systematic runs of K-means for different K, the correct number of clusters can be estimated with the quantitative evaluation of the clustering results as proposed in Section 5.

4 Experiment Data

This experiment was carried out on four two-dimensional datasets and two three-dimensional datasets generated by combining several Gaussian models. Each data point q_i in Q belongs to one of the Gaussian generating models G_j, with the probability

$$p(q_i, G_j) = \frac{1}{(2\pi)^{d/2}|\Sigma_j|^{1/2}} exp[-\frac{1}{2}(q_i - \mu_j)^t \Sigma_j^{-1}(q_i - \mu_j)] \tag{8}$$

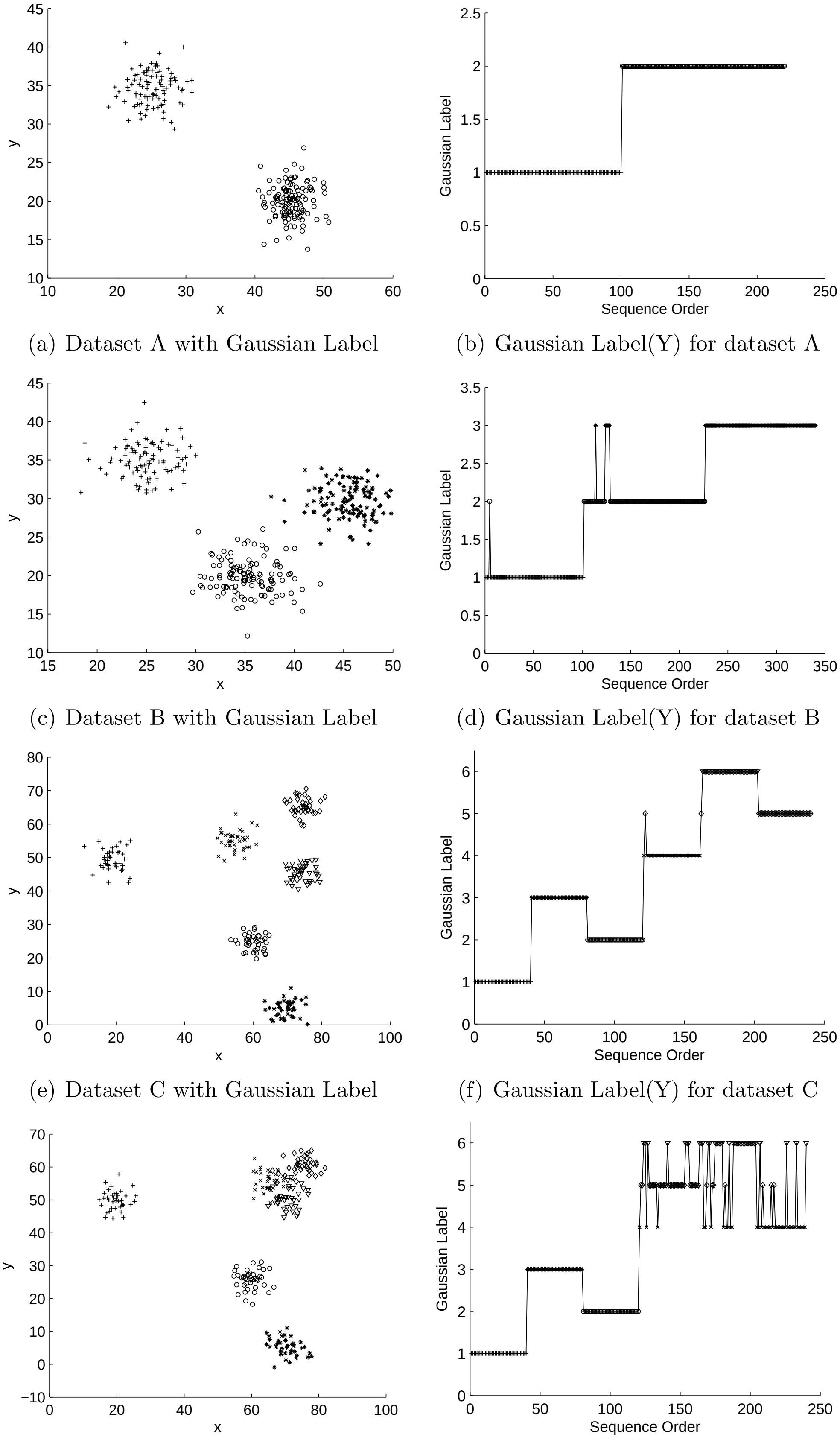
It is natural to consider the generating Gaussian models as the true structure of the clusters. Thus the Gaussian models that generated the dataset are analyzed with the ordering generated by the procedure in Section 3.2. Then K-means was applied for k=2 to 10. For getting a deeper minimum for the variance, the clustering procedure was run 100 times to get the best clustering result with smallest variance in (1). Using the cluster labels, the data points can be compared with the ordered sequence. All analyses in this work are based on the spectrum of cluster labels in a sequence order as shown in Figure 2, where the X-axis corresponds to the ordering, and the Y-axis is the cluster label or Gaussian label. To further verify the performance of K-means, a confusion matrix is adapted to evaluate the clustering result.

4.1 Two-dimensional Dataset

The four datasets shown in figure 2 (a,c,e,g) are generated by Gaussian distribution with parameters as follows:

(a). This dataset is generated by two Gaussian Models with μ_1(25,35),μ_2(45,20), same covariance Σ(5,0;0,5). 100 points are in cluster 1 and 120 in cluster 2.

(b). This dataset is generated by three Gaussian Models with μ_1(25,35), μ_2(35,20), μ_3(45,30), same covariance Σ(5,0;0,5). 120 points for cluster 2 and 3, and 100 for cluster 1.


(a) Dataset A with Gaussian Label

(b) Gaussian Label(Y) for dataset A

(c) Dataset B with Gaussian Label

(d) Gaussian Label(Y) for dataset B

(e) Dataset C with Gaussian Label

(f) Gaussian Label(Y) for dataset C

(g) Dataset D with Gaussian Label

(h) Gaussian Label(Y) for dataset D

Figure 2: (a,c,e,g) data points with Gaussian label shown with marker; (b,d,f,h) Gaussian label plot in sequence order corresponding to dataset ABCD

(c). The dataset is generated by six separable Gaussian models with μ_1(20,50), μ_2 (60,25), μ_3 (70,5), μ_4 (55,55), μ_5 (75,65), μ_6(75,45), same covariance as Σ(8,0;0,8). 40 points for each cluster.

(d). In this dataset, Cluster 1,2,3 are the same as (c), but much closer and some overlap between cluster 4,5,6. μ_4(65,55), μ_5 (75,60), μ_6 (70,50).

It can be seen that clusters in dataset A and C are easily separable, whilesome mixture occurs in B and D. Specifically in D, G_1, G_2, G_3 are well separated while G_4, G_5, G_6 show some overlaps. Each data point is associated with a Gaussian label which indicates the Model that generated it from a different shape marker. Figures 2(b,d,f,h) show the Gaussian labels in the sequence order. (b,d,f) are smooth with few outliers that come from the few points with similar linkage value as pointed out in the figures. But a lot of perturbation or instability occurs in (h) since G_4, G_5, G_6 overlap considerably, and it is hard to separate them correctly. They are much more likely to be one cluster.

4.2 Three-dimensional Dataset

The most attractive feature of this method is that it can order any dimensionality of data points, placing them into a one-dimensional ordering sequence, while preserving clusters' structure. Only three-dimensional datasets are presented here for visualization as an extension of the two-dimensional case. However, the same result can be achieved on data with any dimensionality. Two datasets(E,F) shown in Figure 3 were generated by 3 Gaussian models. Here there is good separation for Dataset E, while two of the three in F are mixed. The corresponding spectrum of Gaussian labels and sequence order is shown on the right side. The mixture area between the two overlapping clusters causes the perturbation on the spectrum.

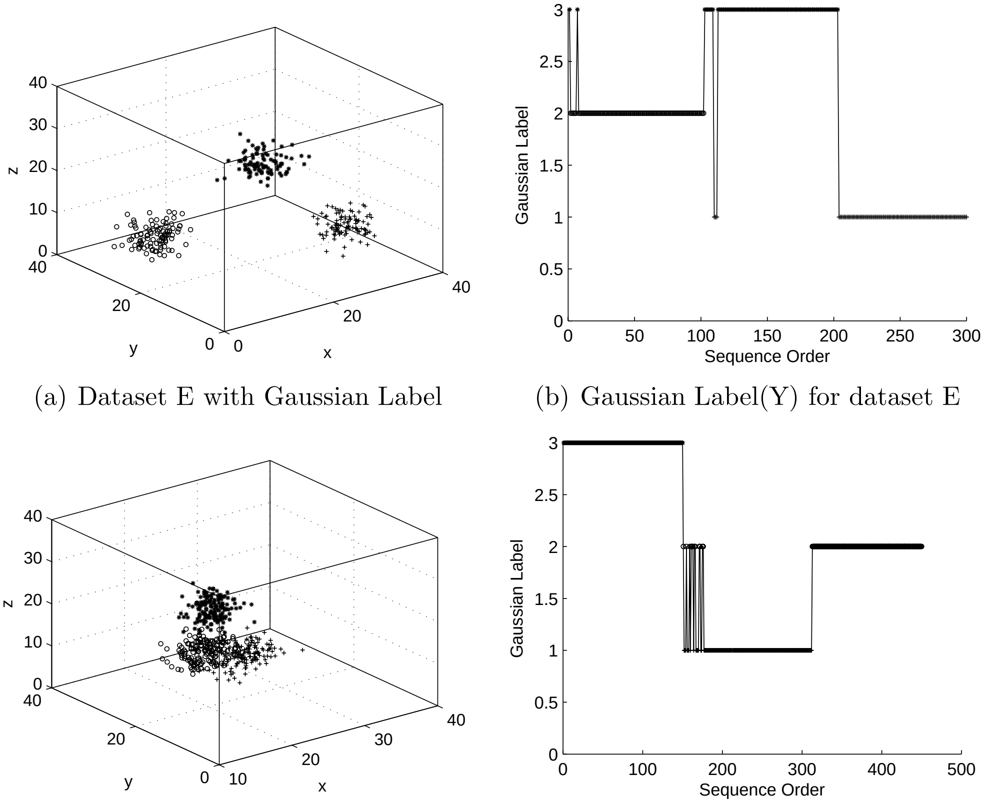
An inappropriate splitting of clusters may lead to many perturbations in the spectrum. Thus the stability of the spectrum can indicate how close the clustering is to a global solution. In the next section, quantitative measures are presented to determine the stability of the spectrum for evaluating the performance of clustering as well as a confusion table for verifying clustering with Gaussian models that generated the data.

5 Spectrum Stability Measurement

5.1 Spectrum Feature Measurement

As discussed above in Section 4, clustering results can be plotted against ordered sequence as a spectrum as shown in Figures 2 and 3. The behavior of the spectrum can suggest the quality or correctness of the clustering. Inother words, correctly clustered points should fall into a continuous interval, rather than jumping back and forth. This section proposes some numerical features to measure the stability of the spectrum.

Let C denote the cluster set, $C = \{c_i, i = 1, .., K\}$. Each data point will be assigned to one of the clusters. For each cluster c_i, I_i is defined as the continued spectrum interval



(a) Dataset E with Gaussian Label

(b) Gaussian Label(Y) for dataset E

(c) Dataset F with Gaussian Label

(d) Gaussian Label(Y) for dataset F

Figure 3: (a,c) data points with Gaussian label shown with marker; (b,d) Gaussian label plot in sequence order corresponding to dataset EF

contains all data points in c_i.

$$I_i = \{m_{i_0}, ..., m_{i_{l-1}}\} \tag{9}$$

with m_{i_0} and $m_{i_{l-1}}$ are the first and the last point assigned in c_i. Feature vector SF containing six elements is proposed for measuring the characteristics of the cluster labels and sequence for each interval of I.

$$SF(I_i) = \{\ell, \alpha, \varphi, \chi, \beta_1, \beta_2\}, i = 1, .., K \tag{10}$$

a)ℓ: Length of the interval

b)α : Occupy rate of the interval by c_i

$$\alpha_i = \frac{n_i}{\ell_i}, n_i \; the \; number \; of \; points \; in \; c_i \tag{11}$$

This feature measures how much data in this interval belongs to c_i . If the clustering is correct,α_i should be larger than incorrect clustering, since n_i=ℓ_i if no other clusters interrupted into this cluster. The larger the α_i is, the purer the interval is.

c)φ and χ :The number of clusters in I_i and which ones as well as how many.

$B_{ij}$is defined to indicate the relationship between clusters c_j and interval I_i:

$$B_{ij} = \begin{cases} 1 & \exists(m \in I_i, m \in c_j) \\ 0 & \text{otherwise} \end{cases} \tag{12}$$

Then,φ and χ can calculated as

$$\varphi_i = \sum_{j=1,..,k} B_{ij} \tag{13}$$

$$\chi_i = \{(c_j, \rho_j, \theta_j) | B_{ij} = 1, j = 1, .., k\} \tag{14}$$

, while ρ_j is the number of points belongs to c_j in interval I_i and $\theta_j = \rho_j / |\ell_i|$ as the percentage of points belongs to c_j in interval I_i. These two show the relationship between different clusters, as well as their structure.

d)β_1 and β_2: Percentage of continued subinterval of I_i belong to c_i Let m_1 and m_2 to be the length of longest and second longest continued subinterval in I_i clustered to c_i. $\beta_{i1} = \frac{m_1}{n_i}$,$\beta_{i2} = \frac{m_2+m_1}{n_i}$ tell the continuity of c_i in I_i . The higher the β_1, β_2 are, the purer and more stable the cluster is.

From the above discussion, good clustering results should have larger α, β_1 ,β_2 with reasonable length but small φ and χ. For each spectrum, the average values of its measurement can be calculated as:

$$(\hat{\alpha}, \hat{\beta_1}, \hat{\beta_2}) = \frac{1}{k} \sum_{j=1}^{k} (\alpha_j, \beta_{j1}, \beta_{j2}) \tag{15}$$

The higher the measurement is, the more stable the spectrum is.

5.2 Confusion Table

For validating the clustering result against a true model, the confusion table CT can be adapted to evaluate the performance.

$$T_{kij} = \begin{cases} 1 & m_k \in G_i, m_k \in c_j \\ 0 & \text{otherwise} \end{cases} \tag{16}$$

$$CT_{ij} = \sum_{k=1}^{n} T_{kij} \tag{17}$$

CT_{ij} is the number of data points that is generated by Gaussian G_i and clustered to C_j . By manipulating the order of cluster labels to get the optimal diagonal sum, the true mapping of Gaussian and cluster can be built. Ratio $R = \frac{trace(CT)}{n}$ is the accuracy of clustering comparing to the generating model.

Now, clustering results, and the ordered sequence and Gaussian models can be analyzed numerically.

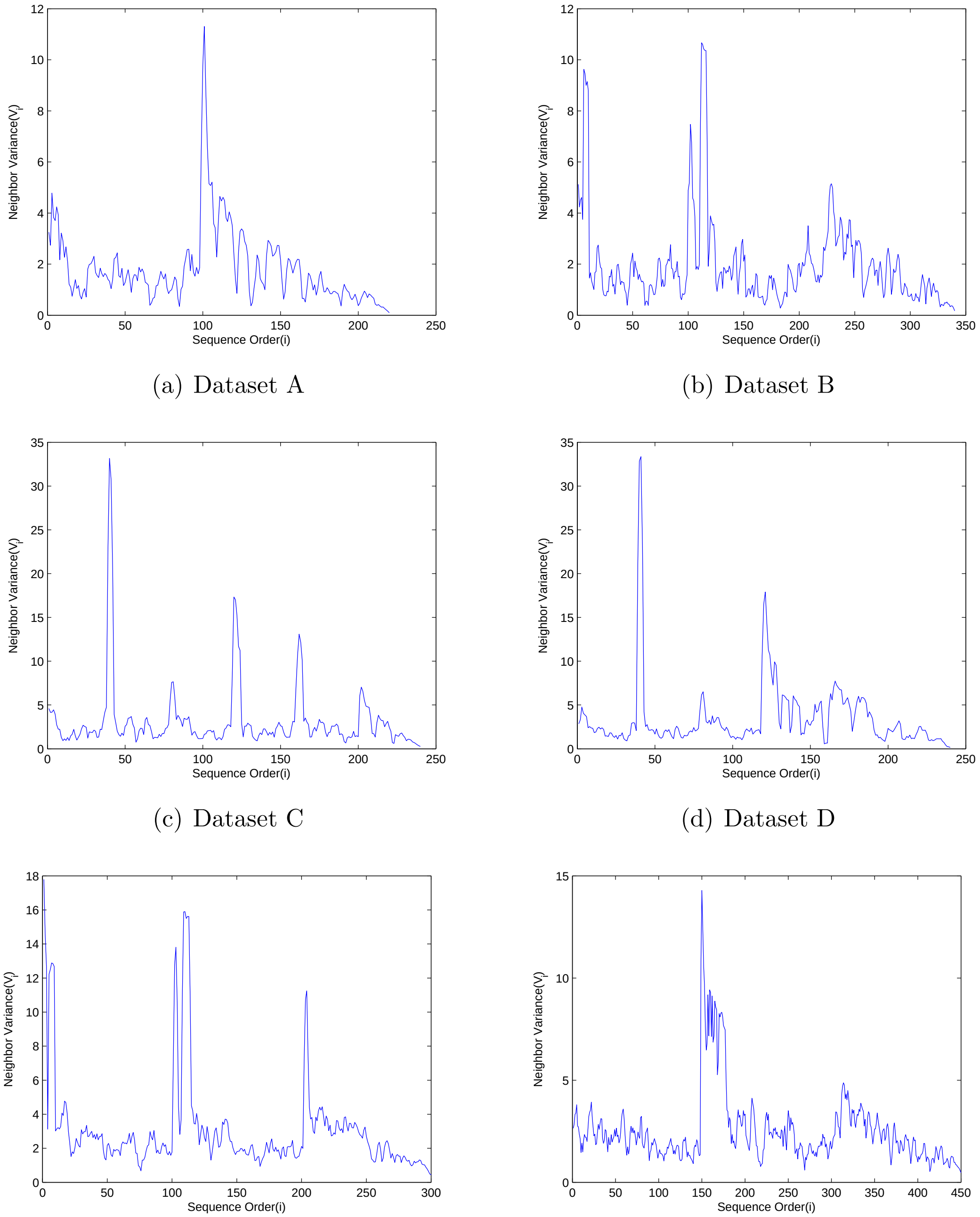
6 Experimental Results

This section discusses the ordering results according to the variance of neighbors for each data point and comparing with K-means clustering on datasets in Section 4 .

As stated in section 2, clusters are ordered in an ordering sequence. Inother words, the sequence visits all points in one cluster and then another. Let $V_i = \frac{\sum_{j \in I}(x_j - \mu_i)}{n+1}$,$I = \{m_{i-n/2}, ..., m_{i+n/2}\}$, V_i is the variance of points in the sequence fragment with length of n+1 centered at point i. V_i is small for points i in dense area, while large when points are in the boundary of clusters. Figure 4 plots the variances of points in length of 5(n=4) for the six dataset discussed in Section 4. From these figures, the structure of these datasets can be clear analyzed and validated with Gaussian models.

The peaks of the variance are the boundaries of cluster intervals. There are two intervals in Dataset A for two clusters, and three for Dataset B. In Dataset C, the figure shows that cluster 1 is far way from others, but cluster 2 and 3 are near each other. The figures shows that the one-dimensional ordering sequence reveals the actual clustering structure of the data in any dimension.

The performances of the K-means procedure for K=2 to 10 are compared with the ordered sequence. The result shows that the clusters are well ordered in the sequence. Good clustering has a more stable spectrum and higher stability value than the visually worse clustering. Results of datasets are presented followed by detailed discussion of the result for dataset D, which is more complicated. Figures(Fig. 5,6,7,8) show comparisons between ordered sequence and K-means clustering for datasets(A,B,C,D). Clustering results for K-means are shown in the left columns with different types of markers indicating the clusters. Corresponding spectrums with cluster labels in the sequence order are given in the right side



(a) Dataset A

(b) Dataset B

(c) Dataset C

(d) Dataset D

(e) Dataset E

(f) Dataset F

Figure 4: Variance of consecutive points in sequence order.

columns. Correct clusterings have stable spectrums. But there is much more perturbation in the spectrums of incorrect clusterings (for incorrect number of clusters K). These figures directly show that the one-dimensional ordering sequence preserves clustering structure for multi-dimensional data.

Data in Figure 5 for two generated groups of points(Dataset A) on a plane shows that the ordering ideally matches the K-means derived clusters for K=2, whereas for K $\geq$ 3 shows a significant perturbation in the ordering. However, the perturbation appears only locally for some clusters like the clustering for K=3 in Fig. 5b which only splits group 1. Data in Fig. 6 for three generated groups(Dataset B) on a plane shows results for K-means clustering with K=2 and 3 with minor perturbation of the ordering spectrum. This small perturbation for K=3 also indicates the small overlap between that two clusters. But for K $\geq$ 4 there is significant perturbations of the ordering, which can be taken as a signal of an incorrect choice of K.

We see from Fig. 7 for the case of six generated groups(Dataset C) on the plane that for K $\leq$ 6 the groupings either have no perturbation of ordering or only slight ones. But for K $\geq$ 7, the signal of ordering perturbation jumps significantly. Fig. 8 illustrates the case(Dataset D) where there are also six generated groups, but three of these have significant overlap so perceptually appear as a single group, for an apparent total four visually perceived clusters. It is clear to find corresponding good clusterings lead to a stable spectrum. For K=2, data points generated by G_2, G_3 are clustered to cluster 2, while data by G_1 and G_4, G_5, G_6 are clustered to cluster 1. The interrupted interval of cluster 1 by cluster 2 is because the ordering sequence starts from G_1(Cluster 1), G_3, G_2(Cluster 2), then G_4, G_5, G_6(Cluster 1). It is interesting to see that our ordering procedure allows that for K-means with K $\leq$ 4 there are no perturbations , as matching the perceptual grouping. For K $\geq$ 5 in contrast, the perturbation of ordering shows that clustering is unstable and will not extract the overlapping and groups reliably, according with the perceptual estimate.

The stability evaluation of the spectrum proposed in Section 5 is applied to evaluate the result of the K-means clustering. The stability value for those clusterings which are correct should be larger than those from incorrect clustering. Tab 1 shows the measurement of the clustering results for K=2 to 6. K-means produce good clustering for K=2 to 4, while there is some jump between c_4 and c_5 for K=5. When K reaches 6, three clusters have an intersection with each other that leads to a small value in α and β_2 . $(\hat{\alpha}, \hat{\beta_1}, \hat{\beta_2})$ in (15) for k=2 to 10 are plot in Figure 9 includes the six datasets. $\hat{\beta_2}$(Blue line) is overcome the $\hat{\beta_1}$(Green line) and always be the larger value when the clustering is good. 2 is the best choice of K for dataset A. Both 2 and 3 are good estimate of K for dataset B with high stability value of $\hat{\beta_2}$, while 6 and 4 for Dataset C and D. The spectrum feature of Dataset E and F shows they can be well clustered into 3 clusters, and dataset F has smaller measurement than E due to its overlapping. The result is exactly matching the perceptual grouping and discussion above.

As discussed above, the measurements show that the clustering for K=4 has higher feature value than the clustering of 6. A confusion table is used to compare the Gaussian generating models (which are the inner structure of the data) with the clustering results. Tab 2 and Tab 3 present the confusion tables for clusters generated by K-means with K=4 and 6. We can

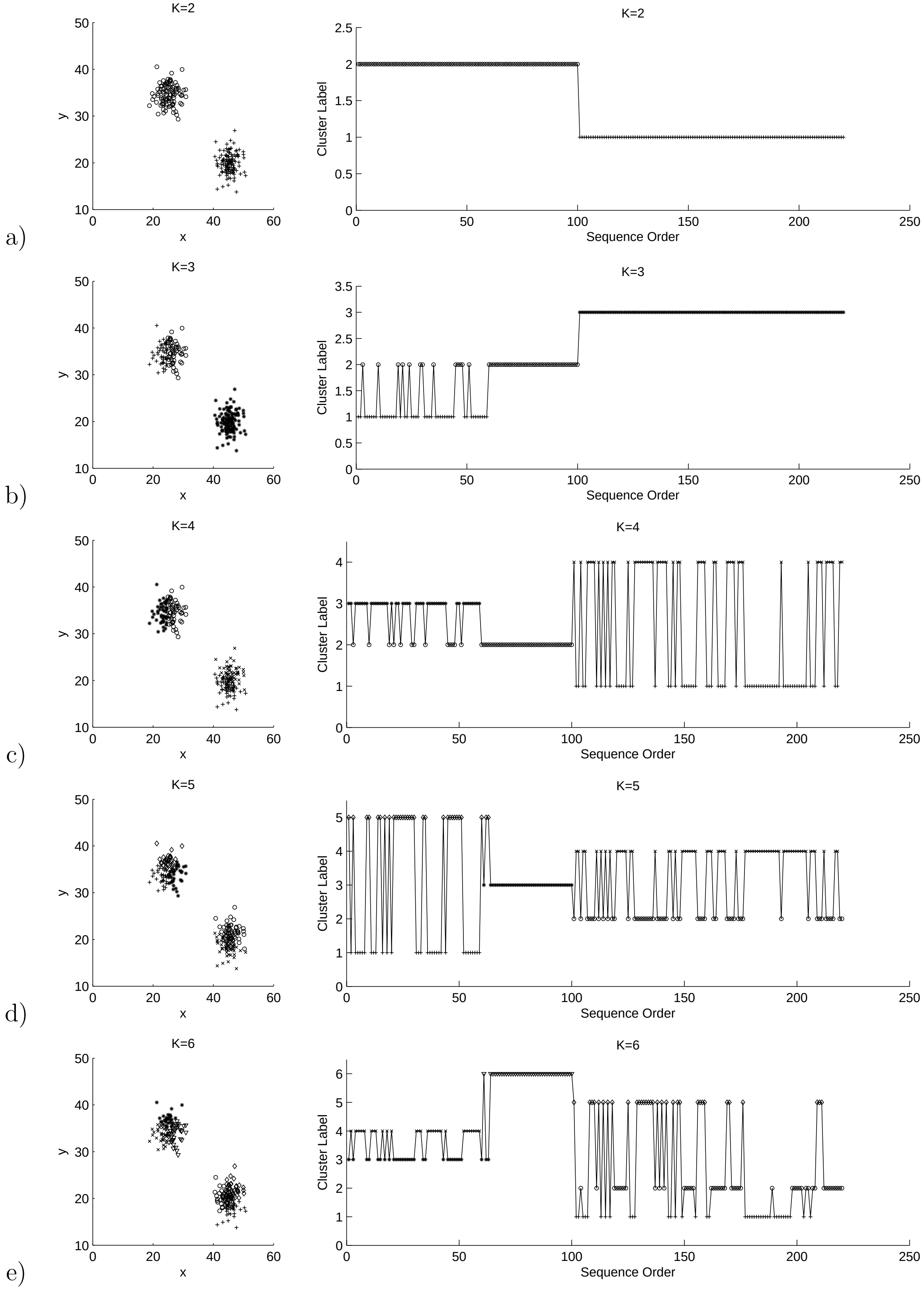


Figure 5: Plot of clustering result of K-means on dataset A with ordering for k=2,..,6

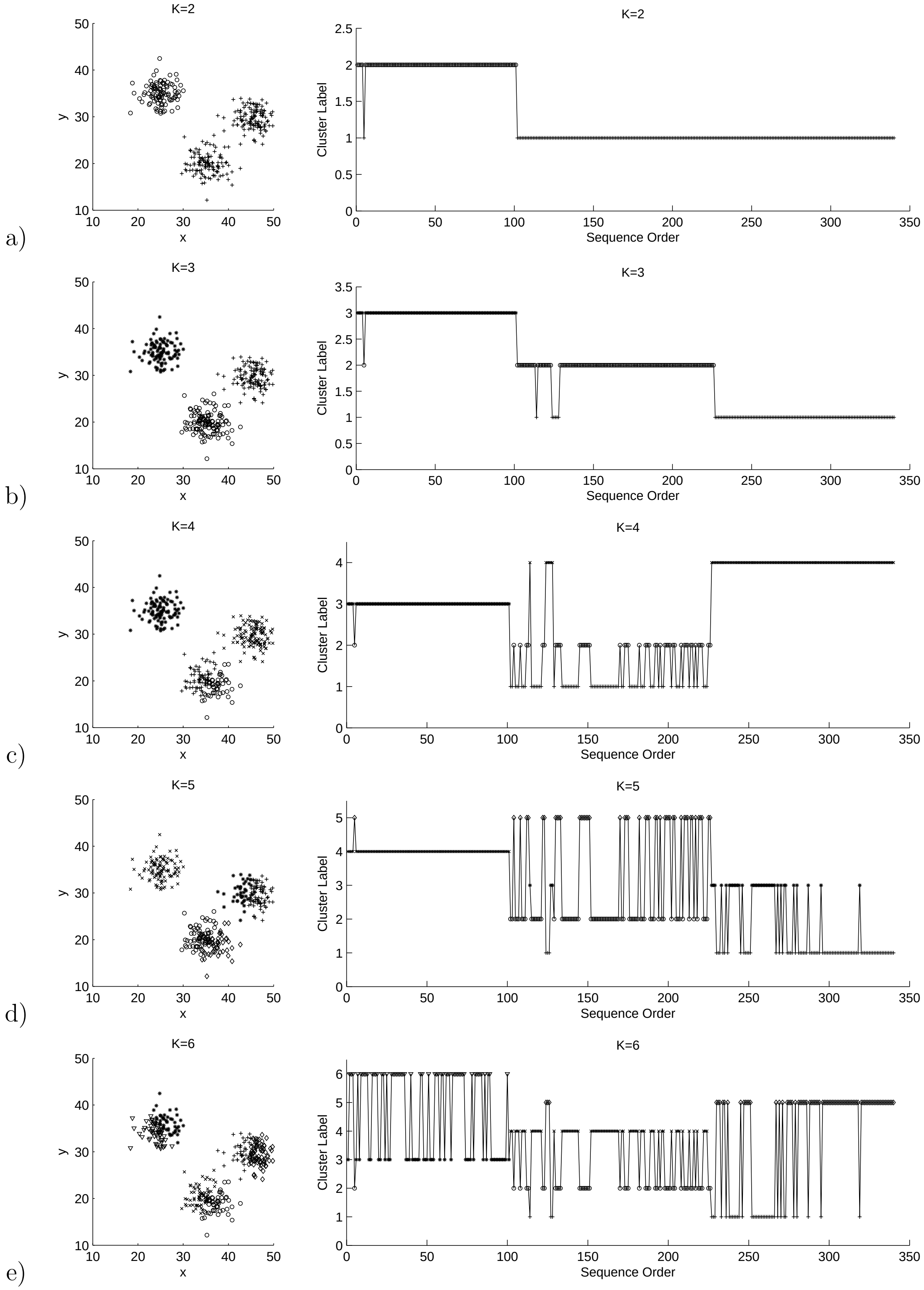


Figure 6: Plot of clustering result of K-means on dataset B with ordering for k=2,..,6

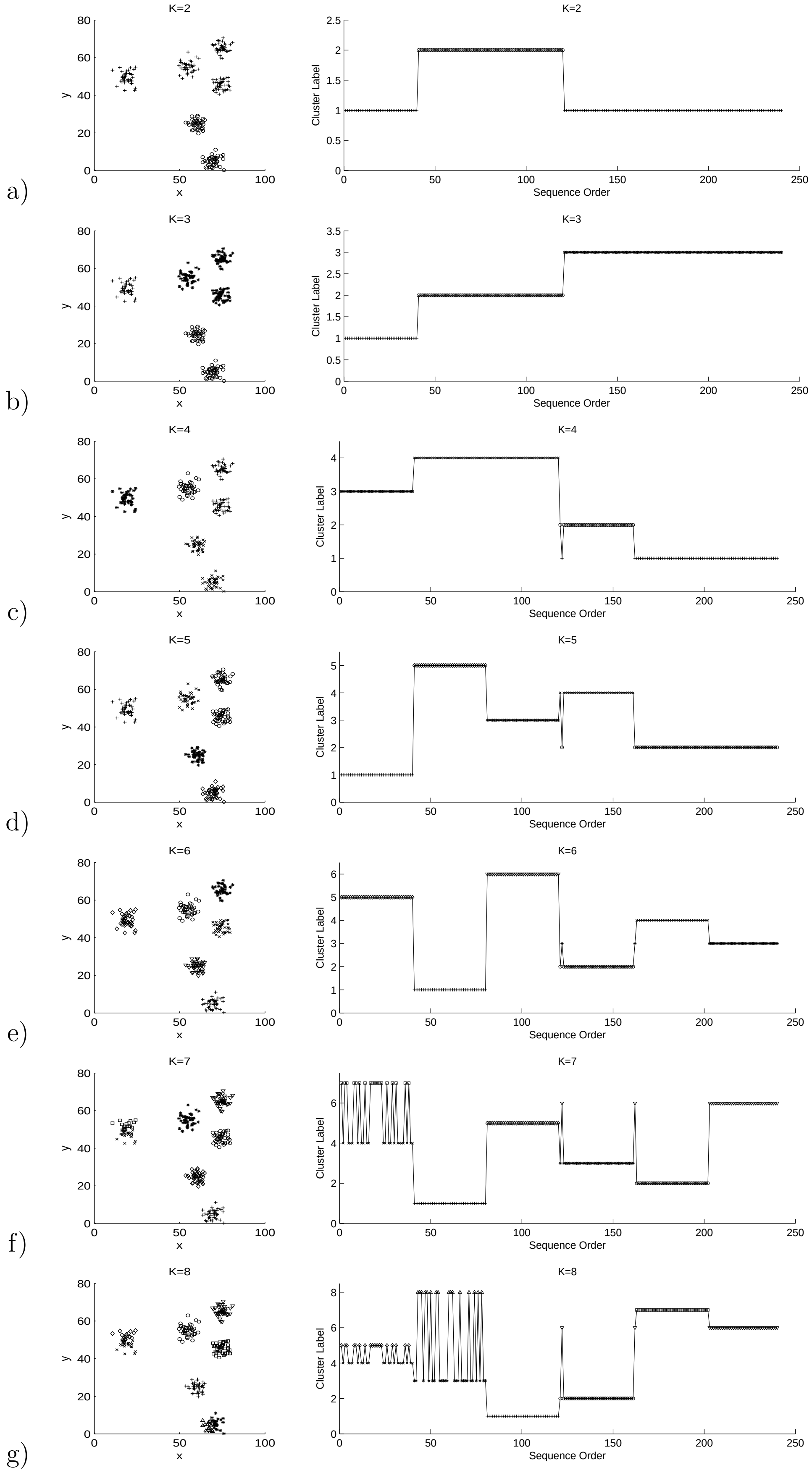


Figure 7: Plot of clustering result of K-means on dataset C with ordering for k=2,..,8

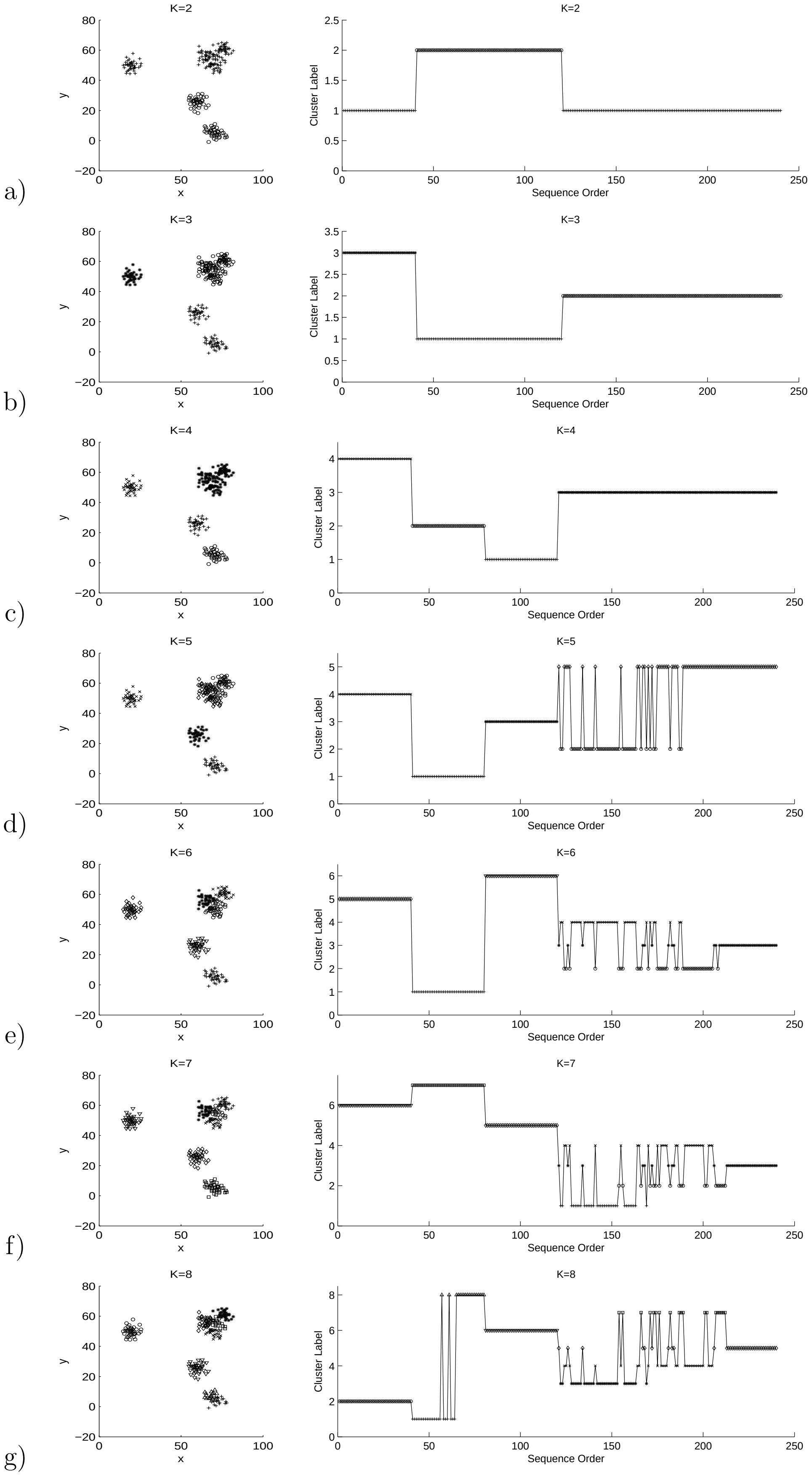
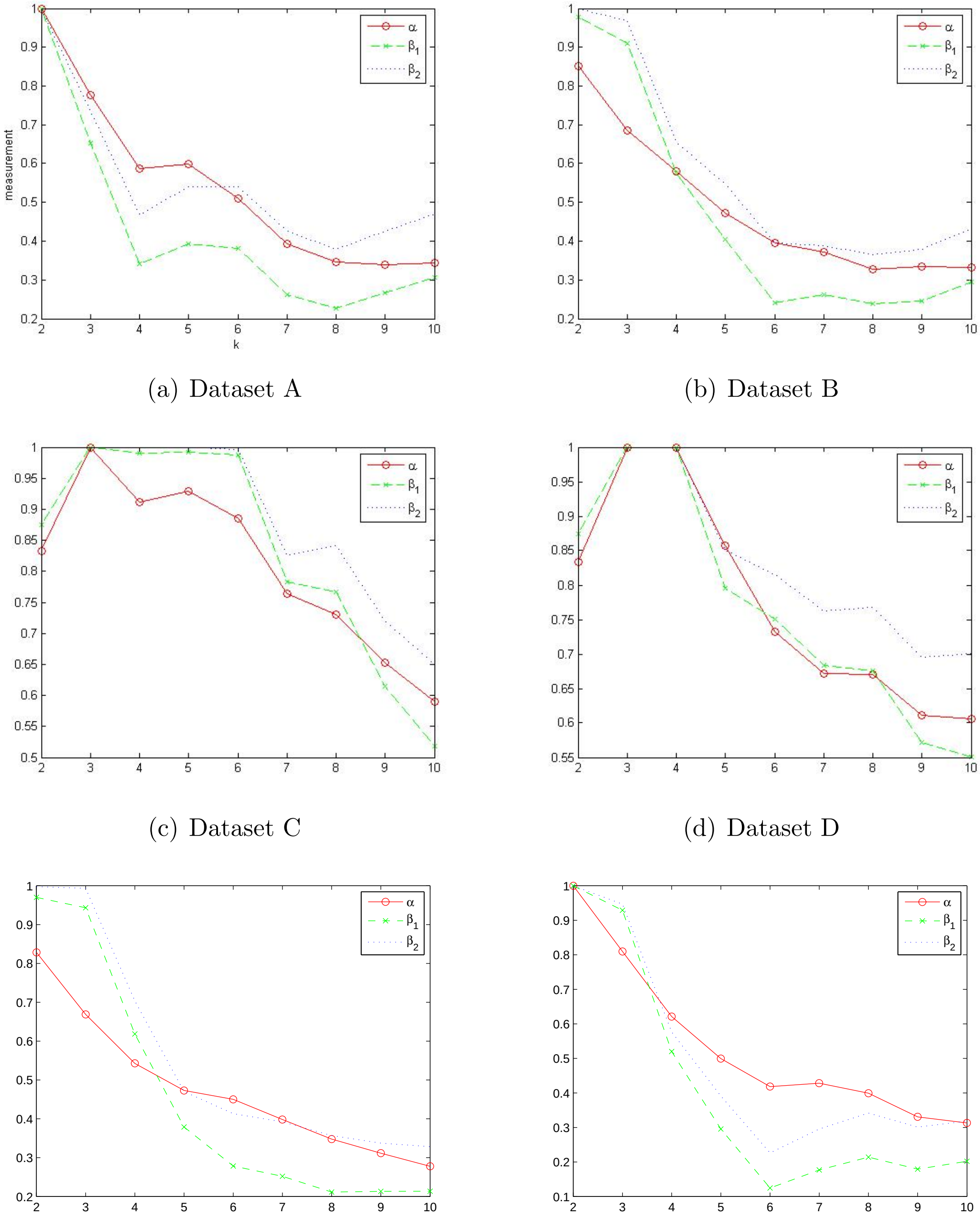


Figure 8: Plot of clustering result of K-means on dataset D with ordering for k=2,..,8



(a) Dataset A

(b) Dataset B

(c) Dataset C

(d) Dataset D

(e) Dataset E

(f) Dataset F

Figure 9: Plot of $(\hat{\alpha}, \hat{\beta}_1, \hat{\beta}_2)$ for k=2,..,10 corresponding to dataset A-F.

k		α	φ	(c,θ)	β_1	β_2
2	c_1	0.667	2	(c_1,0.667),(c_2,0.333)	0.75	1
	c_2	1	1	(c_2,1)	1	1
3	c_1	1	1	(c_1,1)	1	1
	c_2	1	1	(c_2,1)	1	1
	c_3	1	1	(c_3,1)	1	1
4	c_1	1	1	(c_1,1)	1	1
	c_2	1	1	(c_2,1)	1	1
	c_3	1	1	(c_3,1)	1	1
	c_4	1	1	(c_4,1)	1	1
5	c_1	1	1	(c_1,1)	1	1
	c_2	0.642	2	(c_2,0.642),(c_5,0.358)	0.30233	0.488
	c_3	1	1	(c_3,1)	1	1
	c_4	1	2	(c_4,1)	1	1
	c_5	0.642	2	(c_2,0.358) (c_5,0.642)	0.675	0.488
6	c_1	1	1	(c_1,1)	1	1
	c_2	0.435	3	(c_2,0.435),(c_3,0.118),(c_4,0.447)	0.459	0.622
	c_3	0.358	3	(c_2,0.308),(c_3,0.358),(c_4,0.333)	0.744	0.791
	c_4	0.597	3	(c_2,0.284),(c_3,0.119),(c_4,0.597)	0.3	0.475
	c_5	1	1	(c_5,1)	1	1
	c_6	1	1	(c_6,1)	1	1

Table 1: Spectrum Feature Measurement for K-means clustering result on dataset D for k=2,..,6

see from Tab 3 (a) that G_4, G_5, G_6 are clustered to C_4. Let's combine these three Gaussian models. By manipulating the order of Clusters, the maximum diagonal sum of confusion table can achieved. R is 240/240=1 when G_4, G_5, G_6 are combined. After optimizing the confusion table for clusters with k=6, R is 227/240=0.946. 13 points from G_4, G_5, G_6 are clustered incorrectly compared to the Gaussian models. The confusion table result shows that clustering with higher value in stability measurement is better than the smaller one.

7 Conclusion and Future Work

In this work, the relationship between ordering, generating Gaussian models and clustering were analyzed. It has been observed that the clusters of K-means are ordered in the ordered sequence generated by a Quasi-Concave Function with a hierarchical structure for different levels. The numerical measurement of stability can be used to validate the clustering results and measure the performance of a clustering method. If the K-means result is correlated with the clustering presentation given by the ordering produced by our optimization algorithm for Quasi-Concave Set Functions, then it is a good result. By combining the local clustering

(a)

G	c_1	c_2	c_3	c_4
G_1	0	0	0	40
G_2	40	0	0	0
G_3	0	40	0	0
G_4	0	0	40	0
G_5	0	0	40	0
G_6	0	0	40	0

(b)

G	c_4	c_1	c_2	c_3
G_1	40	0	0	0
G_2	0	40	0	0
G_3	0	0	40	0
G'_4	0	0	0	120

Table 2: Confusion table for clustering K=4. (a) Confusion table for six Gaussian model, (b) Confusion table with $G'_4 = G_4, G_5, G_6$.

(a)

G	c_1	c_2	c_3	c_4	c_5	c_6
G_1	0	0	0	0	40	0
G_2	0	0	0	0	0	40
G_3	40	0	0	0	0	0
G_4	0	3	36	1	0	0
G_5	0	0	3	37	0	0
G_6	0	34	4	2	0	0

(b)

G	c_5	c_6	c_1	c_3	c_4	c_2
G_1	40	0	0	0	0	0
G_2	0	40	0	0	0	0
G_3	0	0	40	0	0	0
G_4	0	0	0	36	1	3
G_5	0	0	0	3	37	0
G_6	0	0	0	4	2	34

Table 3: Confusion table for K=6. (a) Confusion table for six Gaussian models, (b) Confusion table after optimizing diagonal sum.

methods with global ordering, a more accurate clustering result can be achieved.

References

[1] S. P. Lloyd, Least squares quantization in pcm, *IEEE Transactions on Information Theory*, 28(2):129136,1982

[2] B. Mirkin, I. Muchnik, Layered Clusters of Tightness Set Functions,' *Applied Math. Leeters*, 15:147-151,2002

[3] Y. Kempner, B. Mirkin, I. Muchnik, Monotone Linkage Clustering and Quasi-Convex Set Functions, *Appl. Math. Letters*, 1997, v. 10, issue no. 4, pp. 19-24

[4] R. Zhang, A. Vashist, I Muchnik, C. Kulikowski, D. Metaxas, A new combinatorial approach to supervised learning: application to gait recognition, *Analysis and modeling of faces and gestures International workshop*, 2005, vol. 3723, pp. 55-69

[5] A. Vashist, C. Kulikowski, I. Muchnik, Ortholog clustering on a multipartite graph, *Workshop on Algorithms in Bioinformatics*, 2005.

[6] T. Le, C. Kulikowski, I. Muchnik, Coring method for clustering a graph, *ICPR*, 2008: 1-4.

[7] R. Duda, P. Hart, D. Stork, Pattern Classification, ISBN:0-471-05669-3, John Wiley & Sons, Inc. 2001